PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F

A2

(11) International Publication Number: WO 99/17182

(43) International Publication Date: 8 April 1999 (08.04.99)

(21) International Application Number: PCT/US98/20627

(22) International Filing Date: 29 September 1998 (29.09.98)

(30) Priority Data:
60/060,628 1 October 1997 (01.10.97) US
09/002,122 31 December 1997 (31.12.97) US

(71) Applicant: CISCO TECHNOLOGY, INC. [US/US]; 170 West Tasman Drive, San Jose, CA 95134–1706 (US).

(72) Inventors: BECHTOLSHEIM, Andreas; P.O. Box 3005, Stanford, CA 94309 (US). CHERITON, David, Ross; 131 Cowper Street, Palo Alto, CA 94301 (US).

(74) Agent: SWERNOFSKY, Steven, A.; The Law Offices of Steven A. Swernofsky, P.O. Box 390013, Mountain View, CA 94039–0013 (US).

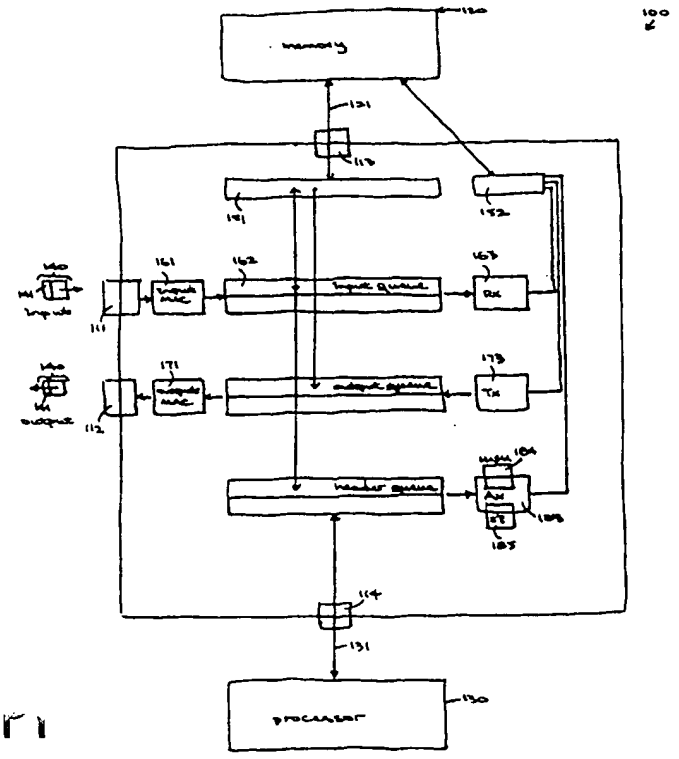
(81) Designated States: CA, CN, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: SINGLE–CHIP ARCHITECTURE FOR SHARED–MEMORY ROUTER

(57) Abstract

The invention provides a single–chip method. The method includes a memory shared among packet buffers for receiving packets, packet buffers for transmitting packets, and packet header buffers for packet forwarding lookup. Accesses to that shared memory are multiplexed and prioritized. Packet reception is performed with relatively high priority, packet transmission is performed with medium priority, and packet forwarding lookup is performed with relatively low priority. The single–chip method includes circuits for serially receiving packet header information, converting that information into a parallel format for transmission to an SRAM for lookup, and queuing input packets for later forwarding at an output port. Similarly, the single–chip method includes circuits for queuing output packets for transmission at an output port, receiving packet forwarding information from the SRAM in a parallel format, and converting packet header information from output packets into a serial format for transmission. The single–chip method also includes a region in its shared memory for a packet forwarding table, and circuits for performing forwarding lookup responsive to packet header information.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

Title of the Invention

Single-Chip Architecture For Shared-Memory Router

This application claims priority of Provisional Application No. 60/060,628, hereby incorporated by reference as if fully set forth herein.

Background of the Invention

1. Field of the Invention

This invention relates to a single-chip architecture for a shared-memory router.

2. *Related Art*

In a packet-switched network, a "router" is a device which receives packets on one or more input interfaces and which outputs those packets on one of a plurality of output interfaces, so as to move those packets within the network from a source device to a destination device. Each packet includes header information which indicates the destination device (and other information), and the router includes routing information which associates an output interface with information about the destination device (possibly with other information). The router can also perform other operations on packets, such as rewriting the packets' headers according to their routing protocol or to reencapsulate the packets from a first routing protocol to a second routing protocol.

It is advantageous for routers to operate as quickly as possible, so that as many packets as possible can be switched in a unit time. Because routers are nearly ubiquitous in packet-switched networks, it is also advantageous for routers to occupy as little space as possible and to be easily integrated into a networking system. For example, implementing a router on a single chip (that is, a single integrated circuit) would be particularly advantageous.

In this regard, one problem which has arisen in the art is that individual integrated circuits and their packages are relatively limited in resources needed to implement a router. In particular, individual chips have only a relatively limited number of pins, a relatively limited die area, and a relatively limited amount of power available for operation. These limitations severely limit the possibility of providing a useful router on a single chip. Routing devices generally need relatively more input and output ports

(thus requiring relatively more pins), relatively more lookup table space (thus requiring relatively larger die size for memory), relatively more packet buffering space (thus requiring relatively larger die size for memory), and relatively more packets routed in unit time (thus requiring relatively larger die size for processing ability and relatively larger power dissipation for speed).

Accordingly, it would be advantageous to provide a single-chip router. This advantage is achieved in an embodiment of the invention in which a router integrated on a single chip shares memory among packet buffers for receiving packets, packet buffers for transmitting packets, and packet header buffers for packet forwarding lookup, and in which accesses to that shared memory are multiplexed and prioritized to maximize throughput and minimize routing latency.

Summary of the Invention

The invention provides a single-chip router. The router includes a memory shared among packet buffers for receiving packets, packet buffers for transmitting packets, and packet header buffers for packet forwarding lookup. Accesses to that shared memory are multiplexed and prioritized. Packet reception is performed with relatively high priority, packet transmission is performed with medium priority, and packet forwarding lookup is performed with relatively low priority.

In a preferred embodiment, the single-chip router includes circuits for serially receiving packet header information, converting that information into a parallel format for transmission to an SRAM for lookup, and queuing input packets for later for-

warding at an output port. Similarly, in a preferred embodiment, the single-chip router includes circuits for queuing output packets for transmission at an output port, receiving packet forwarding information from the SRAM in a parallel format, and converting packet header information from output packets into a serial format for transmission. The single-chip router also includes a region in its shared memory for a packet forwarding table, and circuits for performing forwarding lookup responsive to packet header information.

Brief Description of the Drawings

Figure 1 shows a block diagram of a system including a single-chip router.

Figure 2 shows a process flow diagram of a method for operating a system including a single-chip router.

Detailed Description of the Preferred Embodiment

In the following description, a preferred embodiment of the invention is described with regard to preferred process steps and data structures. Those skilled in the art would recognize after perusal of this application that embodiments of the invention can be implemented using circuits adapted to particular process steps and data structures described herein, and that implementation of the process steps and data structures described herein would not require undue experimentation or further invention.

System Elements

Figure 1 shows a block diagram of a system including a single-chip router.

A system 100 includes a single-chip router 110, a memory 120 coupled to the router 110 using a memory bus 121, and a processor 130 coupled to the router 110 using a processor bus 131.

The router 110 includes a plurality of input ports 111, a plurality of output ports 112, a memory port 113 coupled to the memory bus 121, a processor port 114 coupled to the processor bus 131, and a set of internal memory and internal processing circuits integrated into a single monolithic integrated circuit on at least one side of a silicon die.

In a preferred embodiment, the memory 120 includes an SRAM, and the memory bus 130 includes a 256 bit wide bus operating at about 125 megahertz, so as to provide 32 gigabits per second full duplex communication (that is, both to and from the memory 120). The memory 120 includes sufficient storage to record a set of packets 140 which are received from the input ports 111 and which are in transit to the output ports 112.

The memory port 113 includes a memory data register 151 having 64 eight-bit bytes disposed in a set of four groups of 16 eight-bit bytes, and disposed for receiving data from selected registers of the memory 120 (such as in memory read operations) and for transmitting data to selected registers of the memory 120 (such as in memory

write operations). The memory port 113 also includes a memory address register 152 for selecting the registers of the memory 120 to be read or written.

Packet Receive Circuits

Each one of the input ports 111 is coupled to an input MAC circuit 161, for receiving a set of packets 140 from the input port 111, recognizing a MAC address of the sending device, recognizing a MAC address of the router 110 (as the receiving device), and coupling the packets 140 to an input packet queue 162. In a preferred embodiment, the input MAC circuit 161 receives the packets 140 in a bit serial format and outputs them to the input packet queue 162 as a sequence of eight-bit bytes.

The input packet queue 162 includes a shift register, for receiving the sequence of eight-bit bytes in serial from the input MAC circuit 161, and for transmitting a set of 256 bits (that is, 64 eight-bit bytes) in parallel to the memory data port 151. In a preferred embodiment, the input packet queue 162 is double-buffered; that is, it includes two separate shift registers, one of which can be reading packets 140 in serial from the input MAC circuit 161 while the other can be writing packets 140 in parallel to the memory data port 151.

The input packet queue 162 is coupled to a receive request circuit 163, for determining that the packet 140 has been received (or partially received, if more than 256 bits in length), and for signalling the memory 120 to read the packet 140 from the input packet queue 162. The receive request circuit 163 is coupled to the memory address register 152 and to a control signal for the memory 120.

Packet Transmit Circuits

Similar to the input ports 111, each one of the output ports 112 is coupled to an output MAC circuit 171, for transmitting a set of packets 140 from the output port 112, adding a MAC address for the router 110 (as the sending device), adding a MAC address for the receiving device, and coupling the packets 140 from an output packet queue 172. In a preferred embodiment, the output MAC circuit 161 receives the packets 140 as a sequence of eight-bit bytes and outputs them from the output packet queue 172 in a bit serial format.

Similar to the input packet queue 162, the output packet queue 172 includes a shift register, for receiving a set of 256 bits (that is, 64 eight-bit bytes) in parallel from the memory data port 151, and for transmitting a sequence of eight-bit bytes in serial to the output MAC circuit 171. In a preferred embodiment, the output packet queue 172 is double-buffered; that is, it includes two separate shift registers, one of which can be reading packets 140 in parallel from the memory data port 151 while the other can be writing packets 140 in serial to the output MAC circuit 171.

Similar to the input request circuit 163, the output packet queue 172 is coupled to a transmit request circuit 173, for determining that the packet 140 is ready to be transmitted (or partially ready, if more than 256 bits in length), and for signaling the memory 120 to write the packet 140 to the output packet queue 162. The transmit request circuit 173 is coupled to the memory address register 152 and to a control signal for the memory 120.

Packet Address Lookup Circuits

The input packet queue 162 is also coupled to a packet header queue 182, for isolating a packet header 141 for the packet 140 and for performing address lookup for that packet header 141. In a preferred embodiment, the packet header queue 182 receives the packet header 141 in parallel from the input packet queue 162.

The packet header queue 182 includes a shift register, for receiving a set of 256 bits (that is, 64 eight-bit bytes) in parallel from the input packet queue 162, and for coupling the packet header 141 to an address request circuit 183. In a preferred embodiment, the packet header queue 182 is double-buffered; that is, it includes two separate shift registers, one of which can be reading packet headers 141 in parallel from the input packet queue 162 while the other can be coupling packet headers 141 to the address request circuit 183.

The address request circuit 183 includes a hash circuit 184 for determining a hash address for packet lookup in the memory 120. The hash circuit 184 is coupled to the memory address register 152 for supplying a hash address to the memory 120 for performing packet lookup. The address request circuit 183 is also coupled to a control signal for the memory 120.

In a preferred embodiment, the hash circuit 184 is responsive to a (source, destination) pair in the packet header 141, such as described in detail in the following co-pending patent applications:

- U.S. Application Serial No. 08/581,134, titled "Method For Traffic Management, Traffic Prioritization, Access Control, and Packet Forwarding in a Datagram Computer Network", filed December 29, 1995, in the name of inventors David R. Cheriton and Andreas V. Bechtolsheim, assigned to Cisco Technology, Inc., attorney docket number CIS-019;

- U.S. Application Serial No. 08/655,429, titled "Network Flow Switching and Flow Data Export", filed May 28, 1996, in the name of inventors Darren Kerr and Barry Bruins, and assigned to Cisco Technology, Inc., attorney docket number CIS-016; and

- U.S. Application Serial No. 08/771,438, titled "Network Flow Switching and Flow Data Export", filed December 20, 1996, in the name of inventors Darren Kerr and Barry Bruins, assigned to Cisco Technology, Inc., attorney docket number CIS-017.

These patent applications are collectively referred to herein as the "Netflow Switching Disclosures". Each of these applications is hereby incorporated by reference as if fully set forth herein.

The memory 120 responds to the hash address by delivering a set of packet lookup information to the memory data register 151, which is coupled to the packet header queue 182. The address request circuit 183 also includes a comparator

185 for determining which of several packet lookup responses coupled to the packet header queue 182 is associated with the actual packet header 141.

The packet header queue 182 is also coupled to the processor bus 131, for coupling packet headers 141 and packet lookup information to the processor 130 for extraordinary processing. Thus, when the router 110 is unable to process the packet 140, or processing the packet 140 requires more flexibility than available to the router 110 and the memory 120, the packet header 141 is coupled to the processor 130 for extraordinary processing.

In a preferred embodiment, such extraordinary processing can include enhanced packet forwarding and traffic management services such as access control, multicast packet processing, random early discard, and other known packet processing services.

System Operation

Figure 2 shows a process flow diagram of a method for operating a system including a single-chip router.

A method 200 includes a set of flow points to be noted, and steps to be executed, cooperatively by the system 100, including the router 110, the memory 120, and the processor 130.

At a flow point 210, an incoming packet 140 is received at one of the input ports 111.

At a step 221, the input MAC circuit 161 receives the packet 140 and both recognizes the MAC address for the sending device, and confirms that the MAC address for the receiving device is the router 110.

At a step 222, the input packet queue 162 receives the packet 140.

At a step 223, the receive request circuit 163 determines a location in the memory 120 for the packet 140, and signals the memory 120 to receive the packet 140.

At a step 224, the packet 140 is read into the shared memory 120 from the input packet queue 162.

At a flow point 230, the packet 140 is ready to be routed.

At a step 241, the packet header 141 for the packet 140 is coupled from the input packet queue 162 to the packet header queue 182.

At a step 242, the hash circuit 184 determines a hash address for the (source, destination) pair in the packet header 141, as described in the Netflow Switching Disclosures, hereby incorporated by reference.

At a step 243, the address request circuit 163 couples the packet header 151 to the memory data register 151, couples the hash address to the memory address register 152, and signals the memory 120 to perform a packet address lookup.

At a step 244, the memory 120 performs the packet address lookup and returns its packet lookup results to the memory data register 151. In a preferred embodiment, the memory 120 is disposed as a four-way set-associative memory responsive to the hash address provided by the hash circuit 184, so there are four packet lookup results.

At a step 245, the comparator 185 determines which one of the four packet lookup results is valid for the (source, destination) pair in the packet header 141, and selects that one of the four packet lookup results for packet forwarding.

At a flow point 250, the packet 140 is ready to be transmitted in response to the packet lookup results.

At a step 261, the transmit request circuit 173 determines the location in the memory 120 for the packet 140, and signals the memory 120 to transmit the packet 140.

At a step 262, the packet 140 is read from the shared memory 120 into the transmit packet queue 172.

At a step 263, the output MAC circuit 171 both recognizes the MAC address for the sending device, and adds the MAC address for the receiving device (the router 110 itself), and transmits the outgoing packet 140 on the output port 112.

At a flow point 270, an outgoing packet has been transmitted at one of the output ports 112.

The router 110 operates with regard to each packet 140 using a parallel pipeline. Thus, a first packet 140 is being received while a second packet is being transmitted while a third packet 140 is having a packet lookup performed.

The memory 120 has two regions (a packet buffer region for incoming and outgoing packets 140, and a packet header region for packet header lookup), each of which is intended to be accessed rapidly and often. However, multiple accesses to the memory 120 do not occur simultaneously; instead they are multiplexed so that accesses to these regions are each serviced often by the memory 120, and prioritized so that accesses to these regions can each be serviced rapidly by the router 110.

In a preferred embodiment, packet reception is performed with relatively high priority, packet transmission is performed with medium priority, and packet forwarding lookup is performed with relatively low priority.

Access requests by the receive request circuit 163 have the highest priority, so that when requests for such accesses are received by the memory 120, they are processed before requests for accesses by other circuits. Thus, incoming packets are en-

tered into and retrieved from the input packet queue 162 as quickly as possible, so that queuing at the input ports 111 of the router 110 is minimized.

Access requests by the transmit request circuit 173 have medium priority (after requests by the receive request circuit 163 and before requests by the address request circuit 183), so that when requests for such accesses are received by the memory 120, they are processed after requests for accesses by the receive request circuit 163 and before requests by the address request circuit 183. Thus, outgoing packets are entered into and retrieved from the output packet queue 172 as quickly as possible after incoming packets are processed.

Access requests by the address request circuit 183 have the lowest priority, so that when requests for such accesses are received by the memory 120, they are processed after requests for access by other circuits.

Alternative Embodiments

Although preferred embodiments are disclosed herein, many variations are possible which remain within the concept, scope, and spirit of the invention, and these variations would become clear to those skilled in the art after perusal of this application.

Claims

1. A router, including

at least one port disposed for receiving packets;

at least one port disposed for transmitting packets; and

processing circuits integrated into a single monolithic integrated circuit on at least one side of a silicon die, said processing circuits including

means for accessing a shared memory, said means including (a) circuits disposed for copying packets between at least one region of said shared memory and said processing circuits, and (b) circuits disposed for performing packet lookup in at least one region of said shared memory, said packet lookup being responsive to packet headers of said packets.

2. A router as in claim 1, wherein said at least one port disposed for receiving packets includes a plurality of ports disposed for receiving packets.

3. A router as in claim 1, wherein said at least one port disposed for transmitting packets includes a plurality of ports disposed for transmitting packets.

4. A router as in claim 1, wherein said circuits disposed for copying packets include circuits for receiving packet information in a parallel format and converting that information into a serial format.

5. A router as in claim 1, wherein said circuits disposed for copying packets include circuits for receiving packet information in a serial format and converting that information into a parallel format.

6. A router as in claim 1, wherein said means for accessing said shared memory includes circuits disposed for multiplexing and prioritizing requests for copying incoming packets to said shared memory, requests for copying outgoing packets from said shared memory, and requests for performing packet lookup.

7. A router as in claim 6, wherein said circuits disposed for multiplexing and prioritizing assign relatively high priority to said requests for copying incoming packets to said shared memory.

8. A router as in claim 6, wherein said circuits disposed for multiplexing and prioritizing assign relatively low priority to said requests for copying outgoing packets from said shared memory.

9. A router as in claim 6, wherein said circuits disposed for multiplexing and prioritizing assign priority to said requests in a manner so as to maximize throughput and minimize routing latency.

10. Apparatus including

a shared memory, said shared memory including packet buffers for packets and packet lookup information;

a router coupled to said shared memory, said router including processing circuits integrated into a single monolithic integrated circuit on at least one side of a silicon die, said processing circuits including

means for accessing said shared memory, said means including (a) circuits disposed for copying packets between at least one region of said shared memory and said processing circuits, and (b) circuits disposed for accessing said shared memory for performing packet lookup.

11. Apparatus as in claim 10, including a processor coupled to said router.

12. Apparatus as in claim 10, wherein said means for accessing said shared memory includes circuits disposed for multiplexing and prioritizing requests for copying incoming packets to said shared memory, requests for copying outgoing packets from said shared memory, and requests for performing packet lookup.

13. Apparatus as in claim 12, wherein said circuits disposed for multiplexing and prioritizing assign relatively high priority to said requests for copying incoming packets to said shared memory.

14. Apparatus as in claim 12, wherein said circuits disposed for multiplexing and prioritizing assign relatively low priority to said requests for copying outgoing packets from said shared memory.

15. Apparatus as in claim 12, wherein said circuits disposed for multiplexing and prioritizing assign priority to said requests in a manner so as to maximize throughput and minimize routing latency.

16. Apparatus as in claim 12, wherein said circuits disposed for copying packets include circuits for receiving packet information in a parallel format and converting that information into a serial format.

17. Apparatus as in claim 12, wherein said circuits disposed for copying packets include circuits for receiving packet information in a serial format and converting that information into a parallel format.

18. A method, including the steps of

receiving packets using at least one input port;

transmitting packets using at least one output port; and

routing said packets from said at least one input port to said at least one output port, using processing circuits integrated into a single monolithic integrated circuit on at least one side of a silicon die.

19. A method, including the steps of

receiving packets using at least one input port;

transmitting packets using at least one output port;

copying packets between at least one region of a shared memory and a set of processing circuits integrated into a single monolithic integrated circuit on at least one side of a silicon die; and

performing packet lookup using said shared memory and said processing circuits.

20. A method as in claim 19, wherein said at least one input port includes a plurality of ports disposed for receiving packets.

21. A method as in claim 19, wherein said at least one output port includes a plurality of ports disposed for transmitting packets.

22. A method as in claim 19, wherein said step of copying packets includes the steps of receiving packet information in a parallel format and converting that information into a serial format.

23. A method as in claim 19, wherein said step of copying packets includes the steps of receiving packet information in a serial format and converting that information into a parallel format.

24. A method as in claim 19, wherein said steps of copying packets and performing packet lookup include the steps of multiplexing and prioritizing requests for copying incoming packets to said shared memory, requests for copying outgoing packets from said shared memory, and requests for performing packet lookup.

25. A method as in claim 24, wherein said steps of multiplexing and prioritizing assign relatively high priority to said requests for copying incoming packets to said shared memory.

26. A method as in claim 24, wherein said steps of multiplexing and prioritizing assign relatively low priority to said requests for copying outgoing packets from said shared memory.

27. A method as in claim 24, wherein said steps of multiplexing and prioritizing assign priority to said requests in a manner so as to maximize throughput and minimize routing latency.

28. A method, including the steps of

recording packets and packet lookup information in a shared memory;

coupling processing circuits to said shared memory, said processing circuits being integrated into a single monolithic integrated circuit on at least one side of a silicon die, said processing circuits including (a) circuits disposed for copying packets between at least one region of said shared memory and said processing circuits, and (b) circuits disposed for accessing said shared memory for performing packet lookup.

29. A method, including the steps of

recording packets and packet lookup information in a shared memory;

copying packets between at least one region of said shared memory, and accessing said shared memory to perform packet lookup, using processing circuits which are integrated into a single monolithic integrated circuit on at least one side of a silicon die.

30. A method as in claim 29, including coupling a processor to said processing circuits.

31. A method as in claim 29, wherein said steps of copying packets and accessing said shared memory include the steps of multiplexing and prioritizing requests for copying incoming packets to said shared memory, requests for copying outgoing packets from said shared memory, and requests for performing packet lookup.

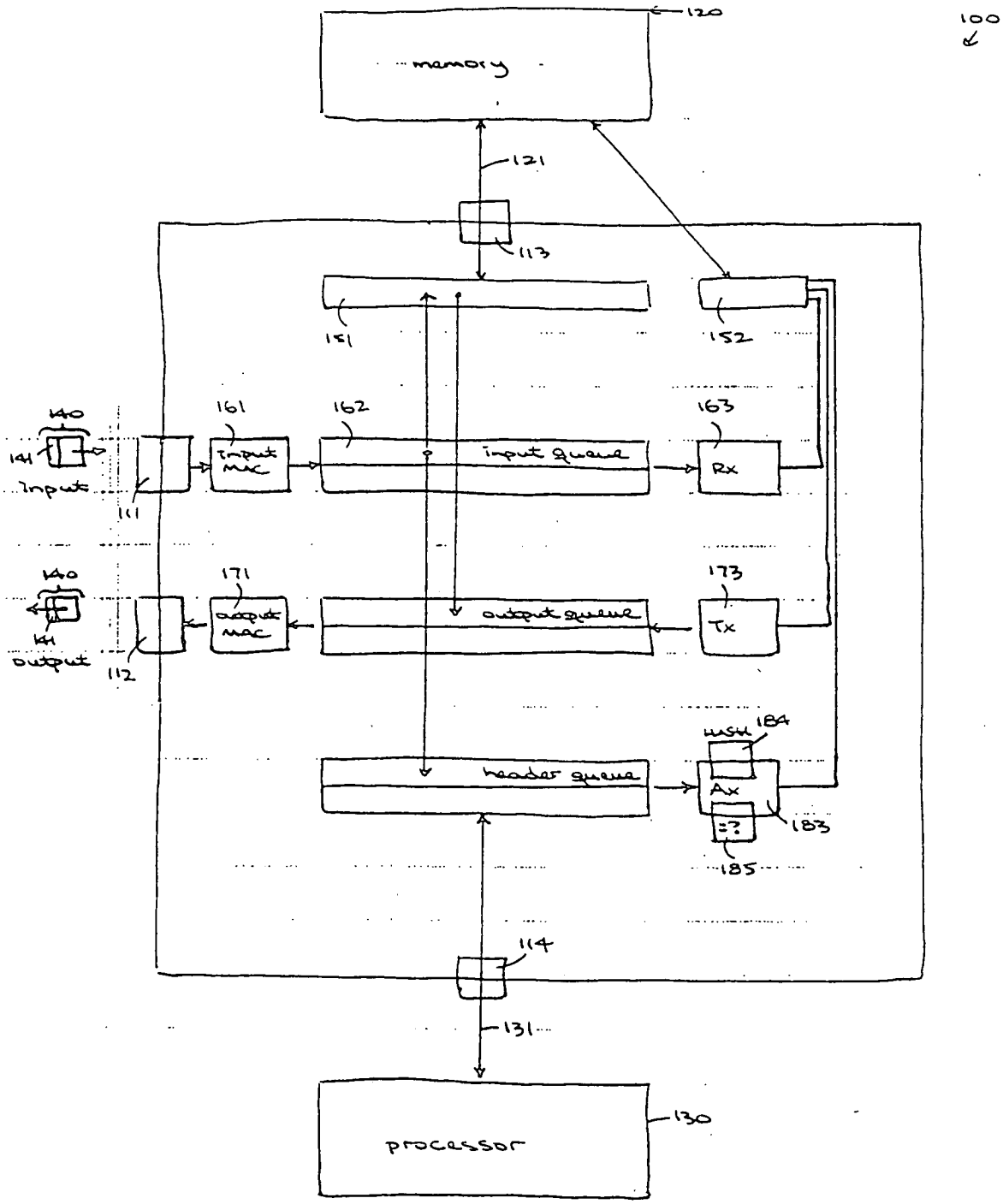
32. A method as in claim 31, wherein said steps of multiplexing and prioritizing assign relatively high priority to said requests for copying incoming packets to said shared memory.

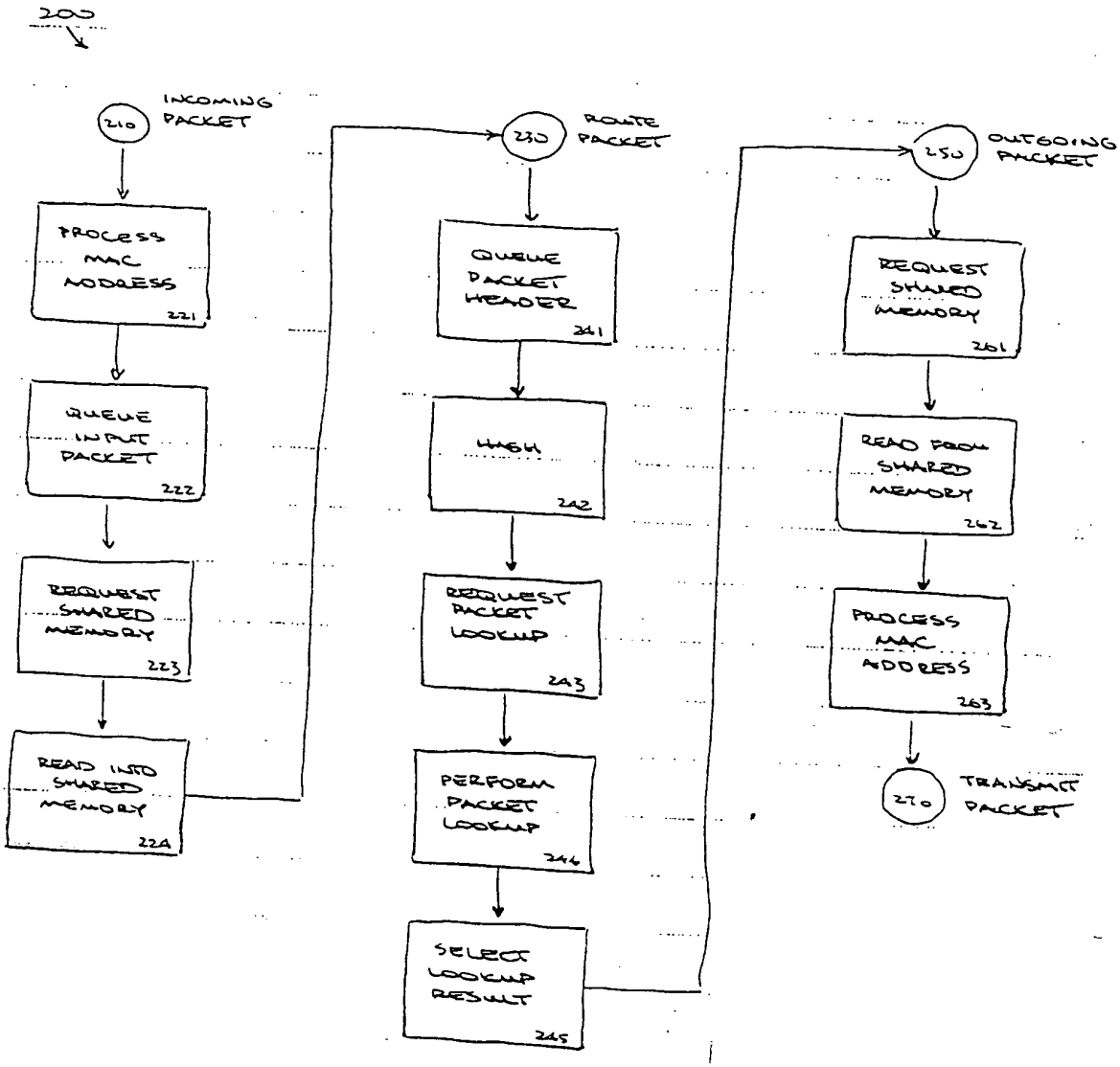
33. A method as in claim 31, wherein said steps of multiplexing and prioritizing assign relatively low priority to said requests for copying outgoing packets from said shared memory.

34. A method as in claim 31, wherein said steps of multiplexing and prioritizing assign priority to said requests in a manner so as to maximize throughput and minimize routing latency.

35. A method as in claim 31, wherein said steps of copying packets include the steps of receiving packet information in a parallel format and converting that information into a serial format.

36. A method as in claim 31, wherein said steps of copying packets include the steps of receiving packet information in a serial format and converting that information into a parallel format.


100
120
memory
121
113
151
152
140
141
Input
111
161
Input MAC
162
input queue
163
Rx
140
141
output
112
171
output MAC
output queue
173
Tx
hash
184
header queue
Ax
183
185
114
131
processor
130


200
210
INCOMING PACKET
PROCESS MAC ADDRESS
221
QUEUE INPUT PACKET
222
REQUEST SHARED MEMORY
223
READ INTO SHARED MEMORY
224
230
ROUTE PACKET
QUEUE PACKET HEADER
241
HASH
242
REQUEST PACKET LOOKUP
243
PERFORM PACKET LOOKUP
244
SELECT LOOKUP RESULT
245
250
OUTGOING PACKET
REQUEST SHARED MEMORY
261
READ FROM SHARED MEMORY
262
PROCESS MAC ADDRESS
263
270
TRANSMIT PACKET